



*Maryland Department of Transportation
State Highway Administration*

Parris N. Glendening
Governor

John D. Porcari
Secretary

Parker F. Williams
Administrator

STATE HIGHWAY ADMINISTRATION

RESEARCH REPORT

CALVERT COUNTY SNOW EMERGENCY DECISION SUPPORT SYSTEM

PHASE II

UNIVERSITY OF MARYLAND

**SP007B4N
FINAL REPORT**

August 2002

The contents of this report reflect the views of the author who is responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Maryland State Highway Administration. This report does not constitute a standard, specification, or regulation.

Technical Report Documentation Page

1. Report No. MD-02-SP007B4N		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Calvert County Snow Emergency Decision Support System Phase II				5. Report Date August 2002	
				6. Performing Organization Code	
7. Author/s Ali Haghani, University of Maryland Masoud Hamedi, University of Maryland				8. Performing Organization Report No.	
9. Performing Organization Name and Address University of Maryland College Park MD 20742				10. Work Unit No. (TRAIS)	
				11. Contract or Grant No. SP007B4N	
12. Sponsoring Organization Name and Address Maryland State Highway Administration Office of Policy & Research 707 North Calvert Street Baltimore MD 21202				13. Type of Report and Period Covered Final Report	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract This goal of this study was to enhance a decision support system for assisting the Maryland State Highway Administration's Office of Maintenance staff in designing snow emergency routes for Calvert County, MD. The research deals with the problem of designing efficient routes for salting and plowing trucks in snow emergencies.					
17. Key Words optimization, snow emergencies, algorithms, routes, plow trucks		18. Distribution Statement: No restrictions This document is available from the Research Division upon request.			
19. Security Classification (of this report) None		20. Security Classification (of this page) None		21. No. Of Pages 68	
				22. Price	

Form DOT F 1700.7 (8-72) Reproduction of form and completed page is authorized.

**CALVERT COUNTY SNOW EMERGENCY DECISION SUPPORT
SYSTEM**

Phase II

Final Report

Submitted to:

The State Highway Administration
Maryland Department of Transportation

Ali Haghani
Professor of Civil Engineering &
Director, Transportation Program
Department of Civil Engineering
University of Maryland
College Park, MD 20742
Phone: (301) 405-1963
Fax: (301) 405-2585
Email: haghani@eng.umd.edu

Masoud Hamedi
Email: masoud@eng.umd.edu

August 2002

Summary and Acknowledgements

This report summarizes the results of research conducted to enhance a decision support system for assisting the Maryland State Highway Administration Office of Maintenance staff in designing snow emergency routes for Calvert County, MD. The research deals with the problem of designing efficient routes for salting and plowing trucks in snow emergencies.

In Phase I the University of Maryland developed a working prototype Decision Support System based on the data available from Calvert County. This system focused on the salting and plowing operations for that county and included a mathematical model and the corresponding solution algorithms for optimizing the salting and plowing operations.

In further discussions with the Office of Maintenance personnel several avenues for enhancement of the prototype system were discussed. Most of the enhancements involved development and implementation of additional capabilities that allow the Office Maintenance greater flexibilities in the use of the system in Calvert County and ultimately in the entire State of Maryland. In Phase II all of these capabilities were added to the system along with some additional ones that were not identified in advance but the Research Team deemed important.

The most important enhancements include a fully interactive graphical user interface, consideration of the constraints that ensure continuity of the truck routes and consideration of multiple depots. The mathematical model that was the optimization tool in the earlier version of the software was re-formulated to incorporate the new constraints. A new solution approach was also developed and implemented in the new version. The new system is fully tested and is ready for use. The test results indicate that the system can design routes that are

operable and provide the same level of service with less number of trucks. The new system also allows the maintenance personnel to design routes manually. The full description of the system capabilities is given in the Users' Manual that is appended to this report. This report focuses on the modeling aspects of the system.

The Research Team wishes to express its sincere appreciation and gratitude toward the Maryland State Highway Administration Office of Maintenance and Office of Research personnel whose help in gathering the necessary data for the project, and whose comments and suggestions was an invaluable asset in the course of the project. Special thanks go to Mr. Russell A. Yurek, Deputy Chief Engineer, Mr. Fran McGrath, the Project Manager, Dr. Richard Woo, Mr. Jeff Smith, and Mr. Rodney Wynn.

TABLE OF CONTENTS

1. Background	1
1.1 Route Design in Snow Emergency Salting and Plowing Operations	1
1.2 The Nature of the Problem	3
1.3 Research Objective and Scope	5
1.4 Organization of the Report	7
2. Literature Review	7
2.1 Arc Partitioning Problem	7
2.2 Phase I Research	8
2.3 Research in Snow Emergency Routing	9
2.4 Arc Routing Problem	12
2.4.1 Chinese Postman Problem (CPP)	12
2.4.2 Capacitated Chinese Postman Problem (CCPP)	12
2.4.3 Rural Postman Problem (RPP)	12
2.5 Capacitated Minimum Spanning Tree Problem (CMST)	13
2.6 Review of the Solution Procedures for the Arc Routing Problem	13
2.6.1 Simple Constructive Methods	14
2.6.2 Two Phase Constructive Methods	16
2.6.3 Improvement Methods	17
3. Problem Formulation and Solution	17
3.1 Set Partitioning Approach	19
3.2 Capacitated Minimal Spanning Tree Approach	20
3.3 The First Fit Heuristic	23
3.4 A Genetic Algorithm with First Fit Heuristic (GAFF)	25
3.5 Routing the Trucks by Finding the Euler Cycle	30
4. Model Implementation for Designing Calvert County Salting Routes	32
5. Conclusions and Directions for Future Research	40
REFERENCES	42
APPENDIX	44

1. Background

1.1 Route Design in Snow Emergency Salting and Plowing Operations

In United States, large amounts of money are invested in snow removal and disposal operation. According to SHRP, snow and ice control operations cost State and local highway agencies in excess of \$1.8 billion annually (1994). Eliminating or reducing snow and ice on the pavements and bridges improve the safety of our highways. However, many planning and routing decisions are not so efficient. Efficient route design and planning for the snow vehicles can not only save highway agencies' operational cost, but also improve level of service and social benefits.

Snow removal and disposal problem is a multi-objective problem. It includes many sub-problems such as which and how much deicer to use, where to locate the depots, how to assign the routes to the vehicles, how to schedule the personnel, etc.

This research focuses on how to assign the routes to the vehicles. A good route assignment can improve the level of efficiency. The efficiency is defined in terms of the amount of time spent treating the roads as a percentage of the total amount of time spent on the route. Figure 1.1 shows the relationship between the time and cost of the operation (Waddell, 1994). Reaction time is the time interval between the weather first beginning to adversely impact the roads and the decision to initiate the snow and ice control operations. Preparation time is the time interval between the decision to start the operations and the beginning of actual road clearing. Execution time is the time interval from the beginning of clearing to the end of the operation. A good routing plan that can reduce the operation time will reduce the operational costs and social costs as shown in Figure 1.2.

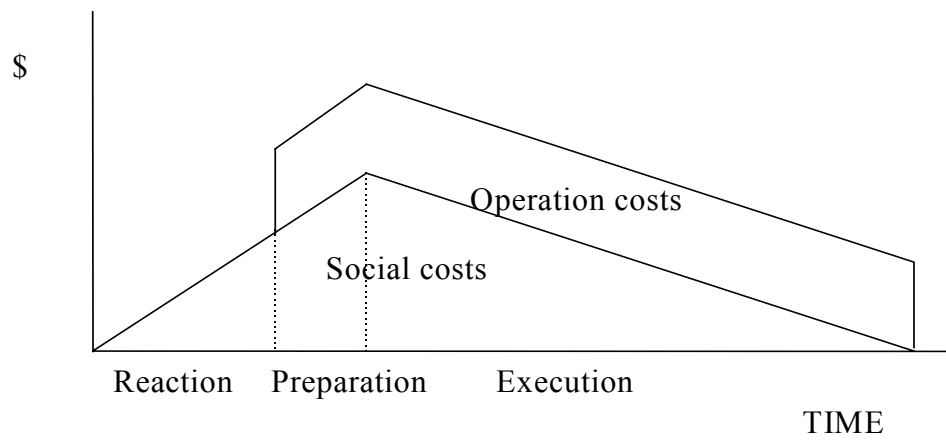


Figure 1.1 Time vs. Cost

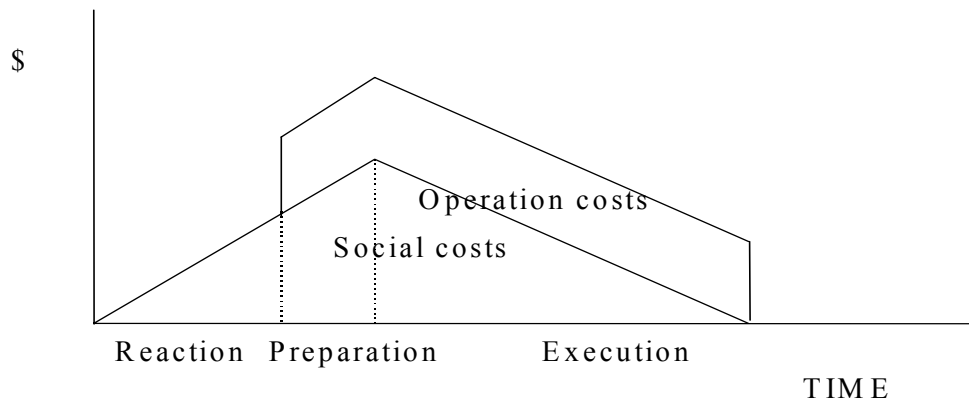


Figure 1.2 Time vs. Cost (Good Route Planning)

1.2 The Nature of the Problem

The objectives of the snow route planning are to find the best route for each vehicle, minimize the total deadhead time and minimize the number of trucks. Snow vehicle routing problems have many features.

First, snow route design can be classified into the category of Arc Routing Problems (ARP) in which the vehicles should service some or all arcs in a network. More precisely, the problem is a Rural Arc Routing Problem in which only a subset of the network arcs needs to be serviced.

Second, in snow emergency operations we are dealing with a directed network in which every link in the network has a direction. In this situation the trucks must service both directions for most roadways. Also, the underlying network is a hierarchical network. This means that there is a precedence structure for servicing the roads. Some roads that are more important should be serviced first. The less important ones can be serviced later.

Golden and Wong (1981) formulated the Capacitated Arc Routing Problem (CARP) that is very close to the snow operation route design problem and suggested many algorithms for that problem. However, a major deficiency of this formulation is that it only deals with non-directed networks. This simplification makes the problem much easier to deal with. Neither Golden's nor any other existing algorithms in the literature can be applied to the capacitated arc routing problem on directed networks, which is a much more difficult problem. Although some researchers suggest approaches to deal with this problem, no mathematical formulation for this problem has been proposed in the literature.

As the name suggests, there are vehicle capacity constraints in this problem. For salting, the trucks can only load a certain amount of salt. When they run out, they have to go back to the depot to load more salt. An important factor that must be considered is that the vehicle capacities are generally different. Many existing approaches assume that the vehicle capacities are the same. In reality, state agencies operate many different types of trucks with different capacities. Therefore for a realistic formulation, we cannot assume that the vehicle capacities are the same. This makes the problem much more difficult.

There may also be some other operational constraints that a truck cannot travel more than a certain amount of time or distance because the driver needs to rest and the truck must refuel.

In general, the route design in snow emergency operations can be described as follows. Given a directed transportation network $G(V, A)$ with arc demands q_{ij} for each arc $(i, j) \in A$, that must be satisfied by a fleet of vehicles k , each of which have a capacity W_k , and starting service time constraints (if any), find the number of cycles which all pass through the domicile and satisfy the demands at minimal total cost.

Several related problems are as follows:

- (1) The Chinese Postman Problem (CPP) that finds a minimum-cost cycle that traverses every arc in the network at least once;
- (2) The Rural Postman Problem (RPP) that finds a minimum-cost cycle that traverses each arc in a given subset of the arcs in the network at least once;
- (3) The Capacitated Chinese Postman Problem (CCPP) in which, given arc demands $q_{ij} > 0$ for each arc (i, j) that must be satisfied by vehicles of capacity W_k , the objective is to

find a set of cycles that all pass through the domicile and satisfy demands at minimal total cost;

- (4) The Traveling Salesman Problem (TSP) that finds a minimum-cost tour that visits each mode in N exactly once;
- (5) The Vehicle Routing Problem (VRP) that finds the minimum cost routes that service all nodes with demand in the network exactly once; and,
- (6) The General Routing Problem (GRP) that finds the minimum cost routes that service a subset of nodes and arcs in the network.

1.3 Research Objectives and Scope

The University of Maryland developed a working prototype Decision Support System based on the data available from Calvert County. This system focused on the salting and plowing operations for that county and included a mathematical model and the corresponding solution algorithms for optimizing the salting and plowing operations.

This system was developed in close coordination with the SHA personnel in Calvert County and the Office of Maintenance. The software system incorporated a major operational constraint on the maximum service route length. The system also included a user-friendly text interface that allowed the SHA personnel to implement changes to these constraints and examine what-if scenarios.

The system was used to examine the existing network of salting operations in Calvert County. The results indicated that a decision support system can assist the maintenance managers in making route assignment decisions and help improve operational efficiency. In further discussions with the Office of Maintenance personnel several avenues for enhancement of the prototype system were discussed. Most of the enhancements involved development and

implementation of additional capabilities that allow greater flexibilities in the use of the system in Calvert County and ultimately in the entire State of Maryland.

The following areas of enhancements were identified:

- (1) An immediate and most important extension to the current system was to develop a graphical user interface for the current system. An interface that allows the users to evaluate what-if scenarios, by just looking at the details of the service networks they generate on a computer screen, was deemed essential to the successful implementation of the prototype system in actual practice. Such an interface can help the user identify the implications of the changes that he or she makes and evaluate the impacts of various operational constraints.
- (2) The following additional capabilities were also deemed necessary:
 - A. Flexibility in adjusting routes.
 - B. Ability to indicate route priorities.
 - C. Ability to handle multiple depots and salt domes.
 - D. Ability to balance the truck drivers' work loads.
 - E. Ability to design salting and plowing routes for a given number of trucks.
 - F. Ability to design continuous routes.

This research focused on enhancing the decision support system to incorporate all of the above capabilities. In the course of the project, the Research Team identified additional capabilities that were not on this list but were added because the Research Team deemed useful. These include the ability to design routes manually, additional safeguards in network information editing, addition of on-line user manual, and a computer simulation of truck routes.

In developing the underlying optimization model we aimed to minimize the total deadhead distance (the distance that the trucks traverse links without salting), and minimize the total number of trucks needed for service.

1.4 Organization of the Report

The rest of this Report is organized as follows. Section 2 provides a comprehensive review of the literature in all related problems. Section 3 presents the mathematical formulation of the problem and discusses the heuristic methods proposed for solving the salting problem and their advantages and disadvantages. In Section 4 the results of the algorithm implementation in Calvert County, Maryland are presented. Finally, Section 5 presents the conclusions.

2. Literature Review

2.1 Arc Partitioning Problem

Snow route design can be classified into the category of Arc Routing Problems (ARP) in which the vehicles should service some or all arcs in a network. More precisely, the problem is a Rural Arc Routing Problem in which only a subset of the network arcs needs to be serviced. In this review we focus on this and other related problems.

The literature on arc partitioning problem is very sparse. In fact, the Research Team did not find any article that directly deals with the problem. There are some papers that, somehow, deal with similar problems. Cowen (1994) presented a polynomial algorithm for network decomposition. The kind of decomposition discussed in the paper is an x -coloring of the nodes of a graph with some properties. This kind of network decomposition is a means of partitioning a network into local regions.

Bodin et. al. (1991) introduced arc partitioning problem that sounds very similar to the approach presented in this report (decomposition of network into connected sub-networks). The arc partitioning problem is a partitioning of a network arcs into a given number of sub-networks so that the workload in each sub-network is as balanced as possible (i.e. is about the same). In this approach the number of sub-networks is given, while in our work, the objective is to minimize the number of sub-networks. A heuristic is presented in the paper that begins by selecting some nodes as seed nodes, then assigns arcs to seed nodes and by interchanging arcs among sub-networks, it tries to balance the sub-networks as much as possible.

2.2 Phase I Research

Phase I report summarized the results of the first stage of the research that was conducted for the MD SHA. Results were obtained that showed promise for improving the operation. There was one limitation in that research. The serviced roads in a route assignment for a truck were not connected to each other. Both the formulation and the algorithm were based on Directed Capacitated Arc Routing Problem (DCARP), while the underlying network was a mixed network as we discussed before.

In that work, we focused on only one goal that was minimizing the total deadhead distance. This is just one of the goals of MD SHA. As mentioned earlier, the other goal is to determine how many trucks are needed to service the area. These two goals are not the same.

We developed a new formulation for the single depot DCARP. The heuristic methods to solve the problem were also suggested. These methods were based on deletion and insertion of links from and into a route. The deletion operation attempts to delete a link serviced by a route while keeping the route connected. The insertion operation attempts to

insert a link into a route with largest savings. Based on these operations, several improvement methods were suggested. These methods include:

1. Modified Augment Method
2. Modified Merge Method
3. Delete and Insert Method
4. Link Exchange Method

All of the above improvement methods were developed for directed networks with a single depot. The final algorithm was a sequential application of these improvement methods.

2.3 Research in Snow Emergency Routing

Dealing with snow emergencies involves a number of different operations. Because each area has its own unique environmental conditions, most papers published for this problem have different operational area backgrounds. For example, Campbell and Langevin (1994) dealt with snow removal and disposal problem for Montreal Urban Community, Cook and Alprin (1976) dealt with the same problem for Tulsa, Oklahoma and Haslam and Wright (1991) studied the problem based on the Indiana state highways. Because different areas have different operations, the formulation and solution approaches can be very different.

The previous literature focuses more on heuristic algorithms instead of finding exact solution algorithms and mathematical formulations. The main reason for this is that the field operations have many constraints that are very hard to describe mathematically. Secondly, even if we are able to formulate the problem mathematically, it is generally very hard to solve as described by Haslam and Wright (1991).

Campbell and Langevin (1994) provided a detailed description of the snow removal and disposal operation in Montreal. They also listed major problems related to snow

operations. These problems include site location, section design, section assignment, fleet mix and routing. An Integer programming formulation for snow disposal site location and assignment problem (SLAP) was given and a decision support system is discussed in that article, but no routing algorithm or formulation was suggested or discussed in the article.

Cook and Alprin (1976) proposed a salt spreader truck routing heuristic algorithm based on closest street selection to balance the travel times to cover all branches in a network. They validated the result with two simulation methods. They listed several advantages of their heuristics:

1. Total time necessary to cover a street network is reduced
2. Total distance covered with salt per unit of time is increased
3. Work load is balanced between trucks
4. Trucks travel more safely over previously salted streets
5. There is automatic coverage in case of truck breakdown
6. Drivers are not required to learn long routes
7. The tedious job of determining routes is eliminated

The advantage of their algorithm is that it is very simple and easily implemented. However, there are some limitations of the heuristic algorithm. First, the snow routes should be divided into appropriate street segments. Second, the truck should be reloaded several times to finish all routes. Third, considering the truck capacity, the nearest routing does not always produce optimal results.

Eglese (1994) presented a heuristic to minimize the distance traveled by gritting vehicles. The solution procedure included partitioning the network, creating the cycle node

network and constructing the route. The heuristic allows multiple depots and different road priorities. Simulated Annealing is used to solve the constrained problem.

Haslam and Wright (1991) discussed the strengths and weaknesses of mathematical programming approaches such as Chinese Postman problem and Multiple Truck Rural Chinese Postman Problem. They suggested a multiple objective heuristic methodology for design of routes for intrastate highway snow and ice control, but reported no detailed result.

Liebling (1973) presented a study for the city of Zurich, based on a CPP procedure and a heuristic to partition the city between vehicles. Marks and Stricker (1971) suggested a cluster-first and route-second heuristic that uses a CPP model for routing. Gilbert (1990) modeled snow blower routing as an asymmetric CPP with duration, precedence and time window constraints and developed an insertion heuristic. Gelinas (1992) described an optimal dynamic programming solution procedure for the CPP with precedence relation and included an application to snow plowing in Montreal. Simulation studies were also conducted to evaluate routes produced by different procedures (England, 1982).

From these studies we can conclude that snow emergency vehicle routing problem is mostly considered as a network optimization problem. More specifically, the routing problem in most snow removal and disposal problems are classified into Capacitated Arc Routing Problems.

Various types of network optimization problems are closely related. Algorithms and the concepts of the algorithm designs developed for one problem, with some modifications, can often be used to solve other problems. A problem can often be transformed, formulated and solved as another problem. Therefore, it is very helpful to review the existing literature in those related problems before we deal with our specific problem.

2.4 Arc Routing Problem

The arc routing problems belong to another subset of the network optimization problems. While in node routing problems we are trying to visit the nodes of the network, in arc routing problems the objective is to traverse the arcs of the network.

2.4.1 Chinese Postman Problem (CPP)

Chinese Postman Problem is simply stated by Guan (1962) as the problem of finding the shortest walking distance for a mailman who has to cover his assigned segment before returning to the post office. Two extensions of the CPP are described as follows. One is windy postman problem (WPP), in which the underlying network is an undirected graph, but the cost of traversing an arc depends on the direction of travel. Another form of the CPP is the hierarchical CPP where a precedence relation is defined on arcs of the graph, and the order in which the arcs are serviced must respect this relation. The CPP can be viewed as the counterpart of the TSP in the category of arc routing problems.

2.4.2 Capacitated Chinese Postman Problem (CCPP)

The capacitated Chinese postman problem is a counterpart of VRP in the arc routing problems and deals with a more realistic case than CPP. Given arc demands $q(i, j) > 0$ for each arc (i, j) which must be satisfied by vehicles with limited capacities, find a set of cycles which all pass through a domicile and satisfy demands at minimal total cost.

2.4.3 Rural Postman Problem (RPP)

While the Chinese postman problem is more likely to arise in urban areas, the RPP is commonly associated with mail delivery in rural areas. There are a number of areas whose set of streets has to be serviced by a postman, and there are other sets of links between those areas

that do not have to be served, but may be used for traveling between those areas. The problem is to find the minimum cost route to service those arcs that must be served.

ARP is a counter part of Node Routing Problem that has field applications. However, it has not attracted as much attention as Node Routing Problem.

2.5 Capacitated Minimum Spanning Tree Problem (CMST)

The minimum spanning tree problem is a fundamental problem in the design of computer communication networks and is extensively used as a basic step in many heuristics that are applied for network design problems. The formulation of this problem can be easily extended to a variety of routing and scheduling problems. Gavish (1982) extended the formulation to different routing problems such as traveling salesman problem and school bus routing problem.

2.6 Review of the Solution Procedures for the Arc Routing Problem

The salting problem is very similar to the Capacitated Arc Routing Problem (CARP). Therefore, in this section, we will focus just on the algorithms for the capacitated problems. Because most arc routing problems are NP-hard, this review will focus more on the heuristic methods. There are few studies that suggest exact algorithms. Recently, Mourao and Almeida (2000) presented two lower-bounding methods, both based on the formulation of a transportation problem, which incorporate some of the side constraints. They also presented a three-phase heuristic to generate a near-optimal solution from the solution obtained with the first lower-bounding method. Another lower bound algorithm is proposed by Benavent, Campos, Corberan and Mota (1992). They use dynamic programming to produce the lower bound.

In his review article, Eiselt (1994) classified the existing heuristics for the capacitated arc routing problem into three categories, 1) simple constructive methods; 2) two-phase constructive methods; and 3) improvement methods.

2.6.1 Simple Constructive Methods

There are five constructive methods that are described as follows:

Construct-Strike Algorithm

This heuristic was first proposed by Christofides (1973) and improved by Pearn (1989). The basic concept of the algorithm is that it gradually constructs feasible cycles and removes them from the network. When a cycle is constructed, the algorithm tries to avoid the disconnected sub-graphs. When feasible cycles can no longer be found, an Euler cycle is constructed on the remaining graph. Pearn modified the algorithm in 1989. He suggested using the minimal spanning tree algorithm to render the remaining graph connected and the matching algorithm to generate an Euler cycle. The construct-strike, minimal spanning tree and matching procedures are repeated until the whole graph is covered.

Path-Scanning Algorithm

The basic idea of this algorithm is to construct one cycle at a time based on a certain optimization criterion. In forming each cycle, a path is extended by joining the arc that looks most promising until the vehicle capacity is exhausted. Then the shortest return path to the depot is followed. Golden, DeArmon and Baker (1983) suggested five criteria:

1. The cost per unit remaining demand is minimized
2. The cost per unit remaining demand is maximized
3. The cost from a node j back to the depot is minimized
4. The cost from a node h back to depot is maximized

5. If the vehicle is less than half full, maximize the cost from a node j back to the depot, otherwise minimize it

Each of these criteria is used to generate a complete solution, and the solution from this approach is the best of these five.

Pearn (1989) modified the path-scanning algorithm. Instead of using each single criterion to generate a complete solution, one of the five criteria is selected randomly to use in each step while extending the cycle path, and generating a complete solution. The problem is solved repeatedly and the best of 30 solutions is then chosen as the best solution for this approach.

Augment-Merge Algorithm

Golden, DeArmon and Baker (1983) suggested an argument-merge algorithm that was inspired by the Clarke and Wright algorithm for the vehicle routing problem (1964). Initially, all arcs belong to different cycles. Then, cycles are gradually merged according to a savings criterion. The basic procedure is outlined below:

- Step 1. Initialize -- all demand arcs are serviced by a separate cycle
- Step 2. Augment -- starting with the largest cycle available, see if a demand arc on a small cycle can be serviced on a larger cycle
- Step 3. Merge -- subject to capacity constraints, evaluate the merging of any two cycles.
Merge the two cycles that yield the largest positive savings.
- Step 4. Iterate -- repeat step3 until finished

The number of augment-merge program executions is limited to 24, and the solution from this approach is the best of the 24 runs.

Parallel Inserting

The parallel inserting algorithm was proposed by Chapleau (1984). It is similar to the path-scanning algorithm but constructs several routes in parallel.

Augment-Insert Algorithm

Pearn (1991) proposed the argument-insert algorithm. The first phase of the algorithm is trying to construct the least cost cycle that covers arcs, and to augment the initial cycle until the vehicle capacity exhausted. The second phase is inserting the remaining arcs into the existing cycles until all capacities are exhausted. The algorithm is specifically used for solving problems with sparse networks and large arc demands.

2.6.2 Two Phase Constructive Methods

Borrowing the ideas from the vehicle routing problem, the two-phase constructive algorithms for CARP can be classified into two categories:

Cluster-First, Route-Second Heuristic

In this algorithm, the arcs are first partitioned into clusters that each has a total weight not exceeding capacity. For this, a greedy criterion can be applied or a generalized assignment algorithm can be used. Then a vehicle route is determined for each cluster by a simple modification of a CPP algorithm

Route-First, Cluster-Second Heuristic

This heuristic first constructs a giant Euler tour over all edges with positive demands. The tour is then partitioned into feasible clusters, this can be done by a bin-packing heuristic, and a vehicle tour is constructed for each cluster.

2.6.3 Improvement Methods

These methods also borrow the ideas from vehicle routing algorithms. While there are some suggestions on the algorithms, there are few implementations of these suggestions. Typically, the improvement methods are inspired by edge exchange heuristics for the traveling salesman problem. Win (1987) suggested the use of simulated annealing. Wang and Wright (1995) suggested using Tabu search to improve the solution. So far no computation results are reported in the literature.

Hertz, Laporte and Mittaz (2000) proposed a Tabu search to solve the problem. On benchmark instances, it outperformed all known heuristics and often produced a proven optimum. De Cordoba, Garcia Raffi and Sanchis (1998) proposed a heuristic algorithm based on Monte Carlo methods. Although the proposed algorithm is for Rural Postman Problem, it can be easily modified to solve CARP. In this algorithm, they simulate a vehicle traveling randomly over the graph, jumping from one node to another on the basis of certain probabilities.

Monte Carlo methods provide a simple approach to many different routing problems. Other Meta-heuristics such as Genetic Algorithm have been applied to solve Rural Postman Problem (RPP). Lee, Kang, and Han (1996) studied chromosome structure, encoding method, decoding method and some operators which are needed when the rural postman problem is solved using their proposed genetic algorithm.

3. Problem Formulation and Solution

The Snow Emergency Routing Problem is simply finding the best routes for snow trucks so that the network is salted as required and the cost is minimized. Each link of the network requires a certain amount of salt and each truck has a given capacity. The problem in

general is a multi-depot arc routing problem. The objective can be either minimizing total deadhead distance (the distance that trucks traverse without salting), or minimizing the number of trucks, or a combination of both.

There is a constraint that makes the problem considered in this research different from other vehicle routing problems and more difficult. This constraint stipulates that the service routes assigned to each truck must be continuous sub-networks. In other word, both directions of a road should be salted by the same truck and a truck cannot serve some part of the network and then go to another part of the network and serve a link that is not connected to the first part. The reason for this route continuity constraint is to avoid confusion for truck drivers.

Formulating Snow Routing problem with route continuity constraint as a mathematical programming problem is very difficult. The problem, therefore, is decomposes into two phases:

1. Assigning segments of network to trucks in such a way that the number of trucks is minimized and all constraints (including the route continuity) are satisfied.
2. Routing trucks in the sub-networks found in step 1.

The second part is simply a rural arc routing problem and is dealt with via existing algorithms in the literature. In the first part we want to partition the arcs of the network into connected sub-networks. Finding the minimum number of trucks to salt a given network while maintaining capacity and route continuity constraints corresponds to the minimal arc partitioning problem in which each partition represents the routes assigned to each truck.

As we discussed before, although there is some literature on the arc partitioning problem, the Research Team couldn't find any article that discusses the Minimal Arc Partition (MAP) problem.

In order to find the optimal partitions of the network we formulate the problem as a set partitioning problem and develop an equivalent Capacitated Minimal Spanning Tree (CMST) formulation as well. Since both formulations are NP-hard we need to resort to heuristics to solve the problem. We propose a First Fit heuristic and a combination of a First Fit heuristic and a Genetic Algorithm for solving the problem.

3.1 Set Partitioning Approach

The minimal arc partitioning (MAP) problem can be stated as: Find the minimal arc partitioning of a network such that:

- Each partition is a connected network,
- Each link in the network belongs to a single partition,
- The union of all partitions constitutes the original network,
- The total service length (link demand) of each partition does not exceed the truck capacity.

Obviously, MAP is a set partitioning problem. Therefore, we can formulate it as:

$$\text{Min} \quad \sum_j x_j$$

$$\text{s.t. :} \quad \sum_j a_{ij} x_j \leq 1 \quad \forall i \in I$$

$$x_j = \begin{cases} 1 & j \in J \\ 0 & \text{otherwise} \end{cases}$$

where:

I : Set of all links

S_k Any subset of the set of links, I . $k = 1, \dots, 2^{|I|}$

\mathcal{S} : $\{\mathcal{S}_k | \text{all links are connected and the total length is not greater than truck capacity}\}$

\mathcal{J} : $\{j | \mathcal{S}_j \in \mathcal{S}\}$

a_{ij} : 1 if link i belongs to set \mathcal{J} and 0 otherwise.

To be able to solve the problem, we need to enumerate all subsets \mathcal{S}_k in \mathcal{S} . This is a time consuming job and adds to the difficulty of solving the set partitioning problem.

3.2 Capacitated Minimal Spanning Tree Approach

We present another approach to formulate the problem in this section. The minimal arc partitioning problem can be transformed into a capacitated minimal spanning tree problem. The transformation is done by converting the original network (the arc network) into another network (the node network) where:

- Each node in the node network corresponds to an arc in the original arc network,
- Each arc in the node network corresponds to a pair of adjacent nodes in the original arc network,
- The flow from each node in the node network correspond to its corresponding arc demand for salt,
- There is a super-node (root) that is connected to all nodes.

The solution of the CMST in the transformed network is equivalent to the solution of the MAP in the original network. The example in Figure 2 illustrates how this works. Through this example, we can see (and it should be easy to prove) that any connected sub-network in the arc network can be represented with a tree rooted at the super node in the node network and vice versa.

If, in the node network, we assign the cost of 1 to all arcs emanating from the root and assign zero costs to all other arcs, a capacitated spanning tree solution to this network is equivalent to the minimal arc partitioning solution in the arc network.

In the example it is obvious that a partitioning of arcs where arcs AE and AB are in one partition corresponds to a tree in the node network that includes nodes R , AE , and AB . Similarly, a tree in the node network that consists of nodes R , AD , DC , and BC corresponds to an arc partition in the arc network that consists of arcs AC , DC , and BC . It can be observed that minimizing the cost in the node network can be interpreted as minimizing the number of arcs connected to the root. This translates into minimizing the number of separated branches rooted at R which finally corresponds to minimizing the number of sub-networks in the arc network. To make sure that the truck capacity constraints hold, the total flow of the nodes of any branch connected to the root should not exceed the truck capacity. Therefore, a CMST solution in the node network will result in a MAP solution in the arc network. The CMST problem can be mathematically formulated as:

$$\begin{aligned}
 & \text{Min } \sum_j x_{0j} \\
 & \text{s.t. :} \\
 & \sum_i y_{ik} - \sum_j y_{kj} = D_k \quad \text{flow conservation at nodes} \\
 & y_{ij} \leq Cx_{ij} \quad \text{truck capacity constraint} \\
 & \sum_i x_{ij} = 1 \quad \text{tree constraint} \\
 & x_{ij} = 0, 1 \quad \begin{cases} 1 & \text{if link } i - j \text{ is used} \\ 0 & \text{otherwise} \end{cases} \\
 & y_{ij} \geq 0
 \end{aligned}$$

The CMST is an NP hard problem that cannot be solved optimally using a greedy algorithm such as Kruskal's or Prim's (Kruskal 1956, Prim 1957). There are some branch and

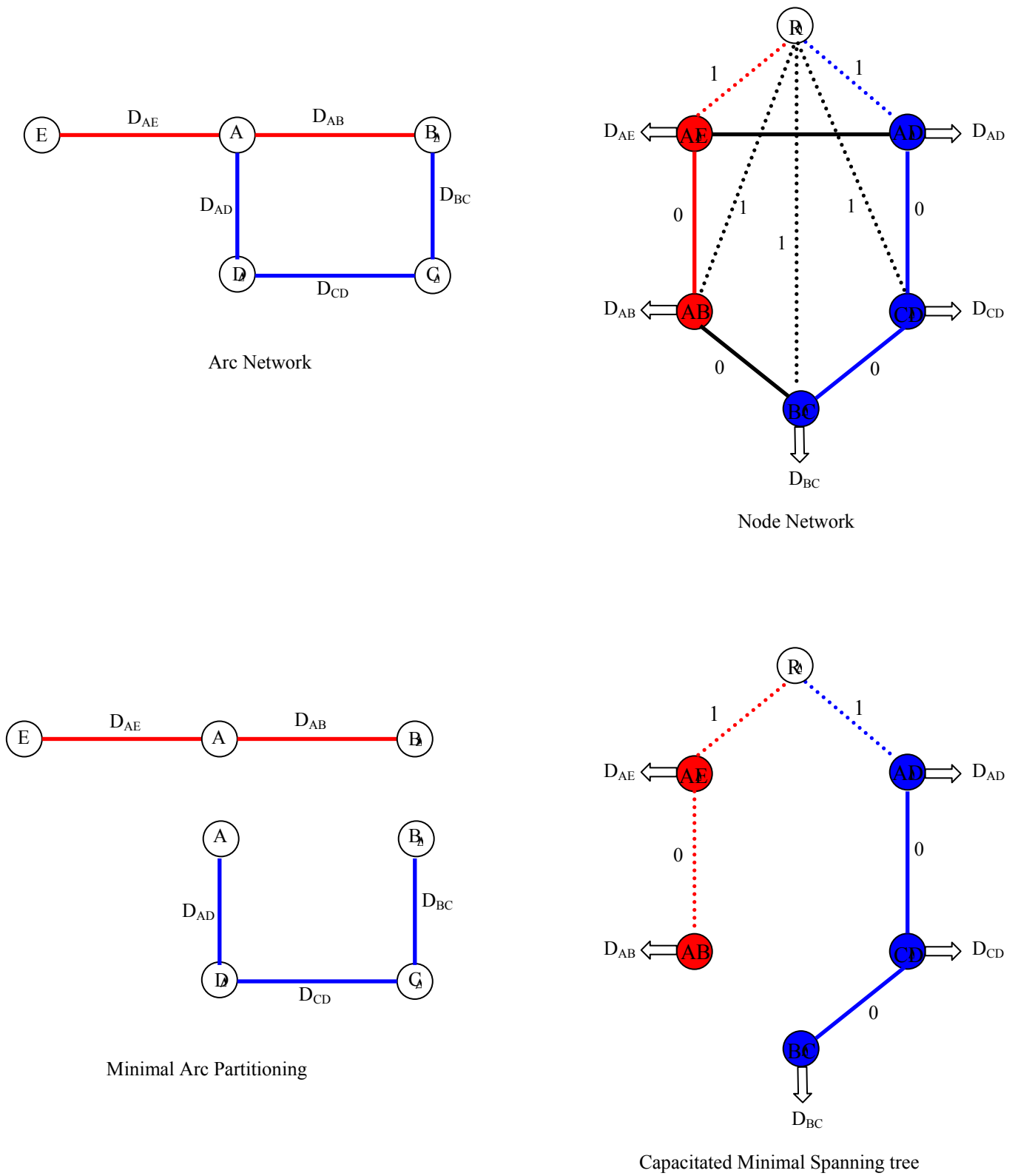


Figure 2
An Example of Transformation of MAP to CMST problem

bound techniques that solve the problem to optimality (Chandy et. al. 1972, Chandy et. al. 1973) and also some heuristic techniques that solve the problem (non-optimally) in a polynomial time.

Kawatra (1998) presented a branch exchange heuristic procedure for solving this problem. He also presented a Lagrangian relaxation method to find a lower bound for the optimal objective function value. Using this lower bound, one may estimate the quality of the solution given by the branch exchange heuristic.

Kershenbaum (1974) checked the efficiency of computing capacitated minimal spanning trees using the branch capacity restrictions, which is a modification of Kruskal's algorithm where weights are assigned to the nodes and then used, along with the arc lengths, to select the order in which arcs are considered for inclusion in the spanning tree.

Gouveia (1994) presented a new formulation for this problem, which is more compact (in terms of the number of constraints) than a well known formulation. Also, he showed that the linear programming relaxations of both formulations produce optimal solutions with the same cost.

In our problem, we don't deal with solving the CMST to optimality. We use the problem to find the lower bound on the number of trucks we need. To do so, we just need to relax the integrality constraint and round up the solution. Applying heuristics to solve the CMST is another approach to solve the problem. Meanwhile, we develop heuristics for the original problem rather than the transformed one.

3.3 The First Fit Heuristic

This heuristic simply constructs a sub-network by adding the first link that fits from the list of links and continue until no more links can be added. The links that fit are those

neighboring the current sub-network and their addition does not violate the truck capacity. When the truck capacity is reached, the sub-network is completed and the next sub-network will be started. The steps for this heuristic are as follows:

- Step 1: Sort the links based on a given key.
- Step 2: Set the current sub-network equal to 1.
- Step 3: Assign the first available link (not yet assigned to any sub-network) to current sub-network.
- Step 4: Find the next available link neighboring the current sub-network and add it to the current sub-network provided that the capacity constraint is not violated. Make the link unavailable. Repeat Step 4 until no more links can be added.
- Step 5: Increment the sub-network number by one.
- Step 6: Repeat steps 3 to 6 until there are no more links available (all links are assigned to sub-networks).

Depending on how links are sorted, different versions of this heuristic can be developed. Two of these heuristics were developed and tested in this project. One is based on increasing demand and the other based on decreasing demand.

In both cases, links are sorted based on the links' demands. The Increasing Demand First Fit heuristic tries to first assign the shorter links to the current sub-network while the Decreasing Demand First Fit does the opposite. This heuristic is a single pass heuristic and, in the worse case, is of the order n^3 where n is the number of links.

3.4 A Genetic Algorithm with First Fit Heuristic (GAFF)

Genetic Algorithm

Genetic Algorithm (GA) is a model of machine learning, which derives its behavior from a metaphor of the processes of evolution in nature. This is done by the creation within a machine of a population of individuals represented by chromosomes, in essence a set of character strings that are analogous to the base-4 chromosomes that we see in our own DNA. The individuals in the population then go through a process of evolution.

Evolution (in nature or anywhere else) is not a purposive or directed process. That is, there is no evidence to support the assertion that the goal of evolution is to produce mankind. Indeed, the processes of nature seem to boil down to different individuals competing for resources in the environment. Some are better than others. Those that are better are more likely to survive and propagate their genetic material.

In nature, we see that the encoding for our genetic information (genome) is done in a way that admits sexual reproduction. Asexual reproduction (such as by budding) typically results in offspring that are genetically identical to the parent. Sexual reproduction allows the creation of genetically radically different offspring that are still having the same general flavor (species).

At the molecular level what occurs (wild oversimplification alert) is that a pair of chromosomes bump into one another, exchange chunks of genetic information and drift apart. This is the *Recombination* operation, which GA generally refers to as *Crossover* because of the way that genetic material crosses over from one chromosome to another. The crossover

operation happens in an environment where the selection of who gets to mate is a function of the *Fitness* of the individual, i.e. how good the individual is at competing in its environment.

Some genetic algorithms use a simple function of the fitness measure to select individuals (probabilistically) to undergo genetic operations such as crossover or *Reproduction* (the propagation of genetic material unaltered). This is fitness-proportionate selection. Other implementations use a model in which certain randomly selected individuals in a subgroup compete and the fittest is selected. This is called tournament selection and is the form of selection we see in nature when stags rut to vie for the privilege of mating with a herd of hinds. The two processes that most contribute to evolution are crossover and fitness based selection/reproduction.

As it turns out, there are mathematical proofs that indicate that the process of fitness proportionate reproduction is, in fact, near optimal in some senses. Mutation also plays a role in this process, though it is not the dominant role that is popularly believed to be the process of evolution, i.e. random mutation and survival of the fittest. It cannot be stressed too strongly that the genetic algorithm (as a simulation of a genetic process) is not a "random search" for a solution to a problem (highly fit individual). The genetic algorithm uses stochastic processes, but the result is distinctly non-random (better than random).

Genetic algorithms are used for a number of different application areas. An example of this would be multidimensional optimization problems in which the character string of the chromosome can be used to encode the values for the different parameters being optimized.

In practice, a genetic model can be implemented by having arrays of bits or characters to represent the chromosomes. Simple bit manipulation operations allow the implementation of crossover, mutation and other operations. Although a substantial amount of research has been performed on variable-length strings and other structures, the majority of work with genetic algorithms is focused on fixed-length character strings.

When the genetic algorithm is implemented it is usually done in a manner that involves the following cycle: Evaluate the fitness of all of the individuals in the population; create a new population by performing operations such as crossover, fitness-proportionate reproduction and mutation on the individuals whose fitness has just been measured; discard the old population and iterate using the new population. Figure 3 depicts a flowchart for the genetic programming paradigm.

We have developed a heuristic based on the Genetic Algorithm for the problem. The evolution program is a probabilistic algorithm, which maintains a population of individuals at each iteration. Each individual represents a potential solution to the problem. Each individual is evaluated to give a measure of its fitness. Then, a new population is formed by selecting the more fit individuals (select step).

Some members of the new population undergo transformations by means of genetic operators to form new solutions. There are unary transformations (mutation type), which create a new individual by a small change in a single individual. And there are higher order transformations (crossover type), which create new individuals by combining parts from several (two or more) individuals. After some number of generations, the program converges.

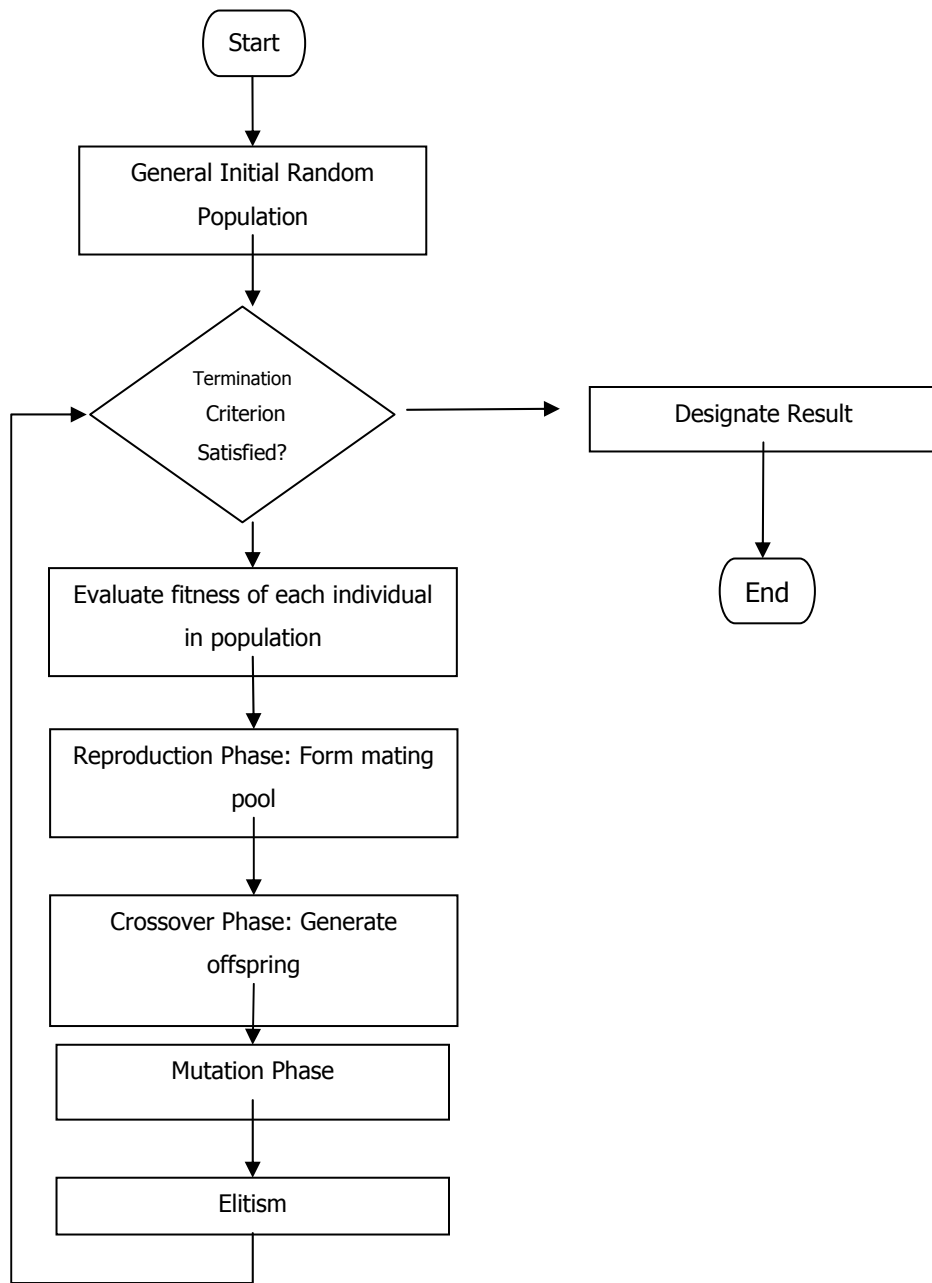


Figure 3
Flow Chart of a Typical Genetic Algorithm

It is hoped that the best individual represents a near-optimum (reasonable) solution (Gen et al., 2000; Michalewicz, 1999).

The Genetic Algorithm in this project uses random key method for representation and initialization of chromosomes. To evaluate chromosomes, the First Fit heuristic is applied. The selection phase is based on roulette wheel selection that gives more reproductive chances to those population members that are the fit.

The idea of elitism is also incorporated in the algorithm. Elitism simply says that it is probable that the best member of a population does not produce offspring in the next generation. So, this strategy, copies the best member into succeeding generation.

The reproduction phase is based on two genetic operators: two point crossover and mutation. The two-point crossover swaps two randomly selected chromosomes at randomly selected points and mutation randomly changes the value of selected genes.

The steps for the GA heuristic developed to solve the problem are as follows:

- Step 1: Generate initial population based on random keys method.
- Step 2: Generate routes using First Fit Heuristic and the random keys.
- Step 3: Evaluate the population and update the best-ever-found solution.
- Step 4: Apply Roulette Wheel selection method to generate the new population (Selection).
- Step 5: Randomly replace a chromosome with the best ever found solution (Elitism).
- Step 6: Apply two point crossover based on crossover probability (Crossover).
- Step 7: Based on mutation probability, select and replace genes with random numbers between 0 and 1 (Mutation).
- Step 8: Repeat steps 2 to 7 until the convergence criteria is met.

3.5 Routing the Trucks by Finding the Euler Cycle

After determining the service sub-network for each truck, we need to determine the route that the truck has to traverse to perform the service. This problem is solved by finding an Euler Circuit in each of the sub-networks (Larson and Odoni, 1981).

Let G be a connected graph (every pair of nodes is joined by some path). An *Euler path* is a path using every edge of the graph G exactly once. An *Euler circuit* is an Euler path that returns to its start: a closed path. The necessary and sufficient condition for G to have an Euler circuit is that each node of G has even degree. We can construct an Euler circuit in the following way.

1. Start at some node a .
2. Leave a along one of the edges adjacent to it, and continue on a path, always leaving a node by a different edge than the one you came in on, until you arrive back at a .
3. If all edges are used, then an Euler circuit is obtained.
4. Otherwise, remove all edges of the circuit just traversed from the graph G . Also remove all nodes whose only edges are on that path.
5. The result is G_1 , a sub-graph of G .
6. Notice these facts:

Every node in G_1 has even degree. This is because an even number of edges were removed from each node, namely the edges belonging to the path just completed.

There is some node b that is common to the completed path and graph $G1$. In other words, not all the edges of b were on the completed path. Otherwise, b would have removed when the edges of that path were removed. But *some* edges of b were on the path.

7. Start at node b and repeat steps 1-4. If this still does not exhaust all edges in the graph, then repeat.
8. Since there are finitely many edges in G , this process has to end. At that point a finite collection of circuits are obtained and in each circuit each edge will have been used no more than once.
9. Now the circuits can be glued together using the common nodes (like b). Each such node is a common node between a circuit already completed and the next one being constructed.
10. This gives an Euler circuit. The circuit can be constructed by starting at node a . When common node b between the first circuit and the second one is reached, continue on through the next circuit until *its* common node c is reached (if any). When returning from a common node, the unused portion of each circuit will be traveled to its starting point. At the end, a will be reached after traversing each edge exactly once.

We use the above method to route the trucks in the sub-networks that represent their service areas.

4. Model Implementation for Designing Calvert County Salting Route

In Calvert County the salting routes are divided into four sections. In this research we will consider the routes in all four sections of the salting network. We use the algorithm proposed in this research to design the route structure for these sections and compare our results with the current route structure that is in place in the county. Currently, there are fourteen trucks servicing this area and all of them have the same salt carrying capacity of 16,000 lbs (8 tons).

The heuristics are implemented in Visual C++ and are embedded in the enhanced decision support system. The system is used to design snow emergency routes for Calvert County. The underlying network consists of 42 nodes and 52 arcs grouped into 4 sections. Figure 3 shows the underlying network.

To demonstrate truck routes, a graphical user interface is developed by the Research Team. Figure 4 shows the output of this interface for all truck routes for the current route assignment. Different colors represent different trucks and their corresponding routes. Figure 5 shows the route assignment for a single truck.

The Genetic Algorithm with First Fit heuristic produces excellent results. Table 1 shows the details of the routes produced by the system. Table 2 shows the summary of the route information produced by the system. The minimum number of trucks obtained in this approach is 12 showing a reduction of 2 trucks or %14. The solution time is very low and the convergence is very quick. The final result of 12 trucks was obtained at the 6th iteration and the result remained unchanged up to 30th iteration.

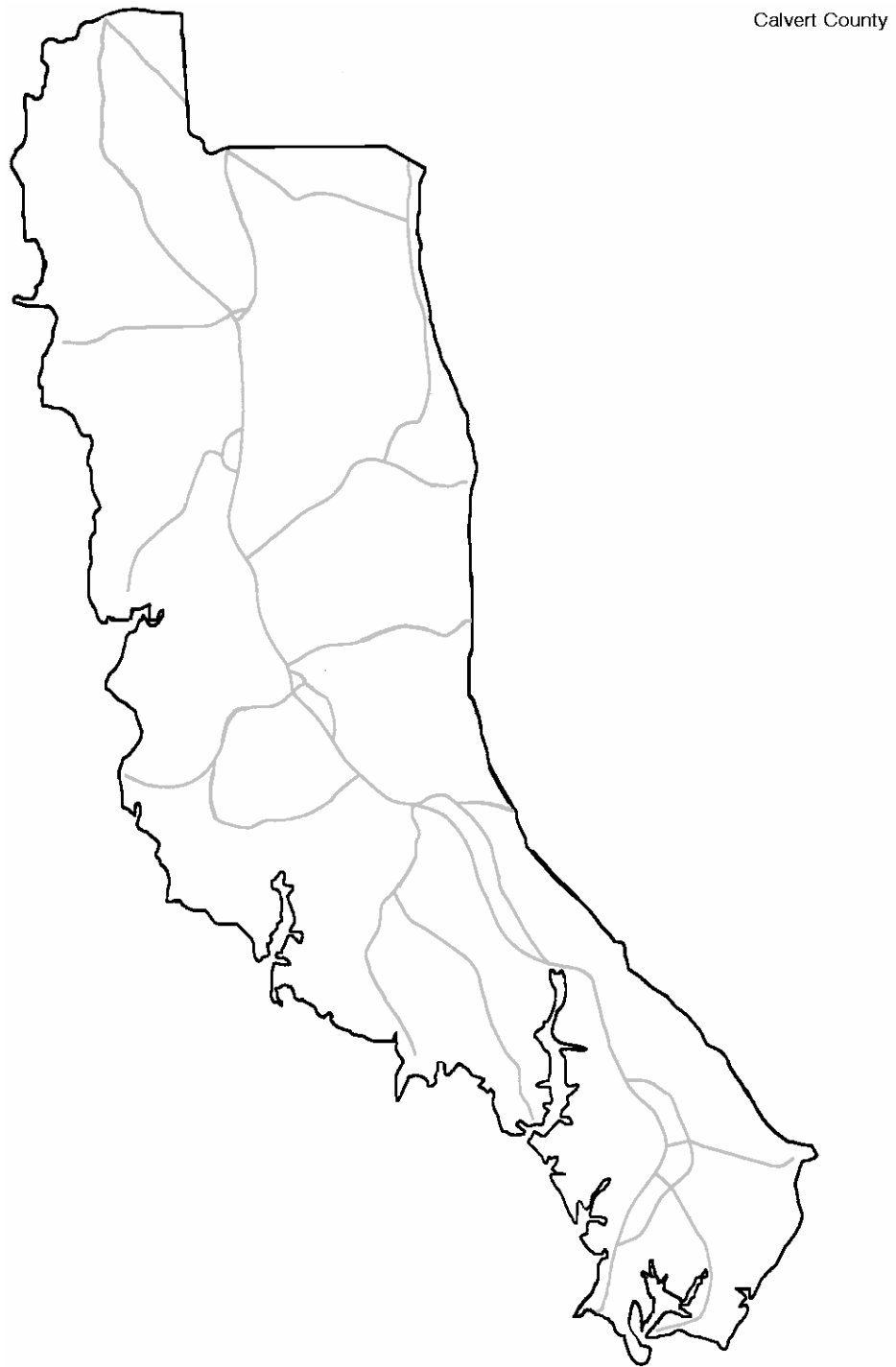


Figure 3

Calvert County Snow Emergency Network

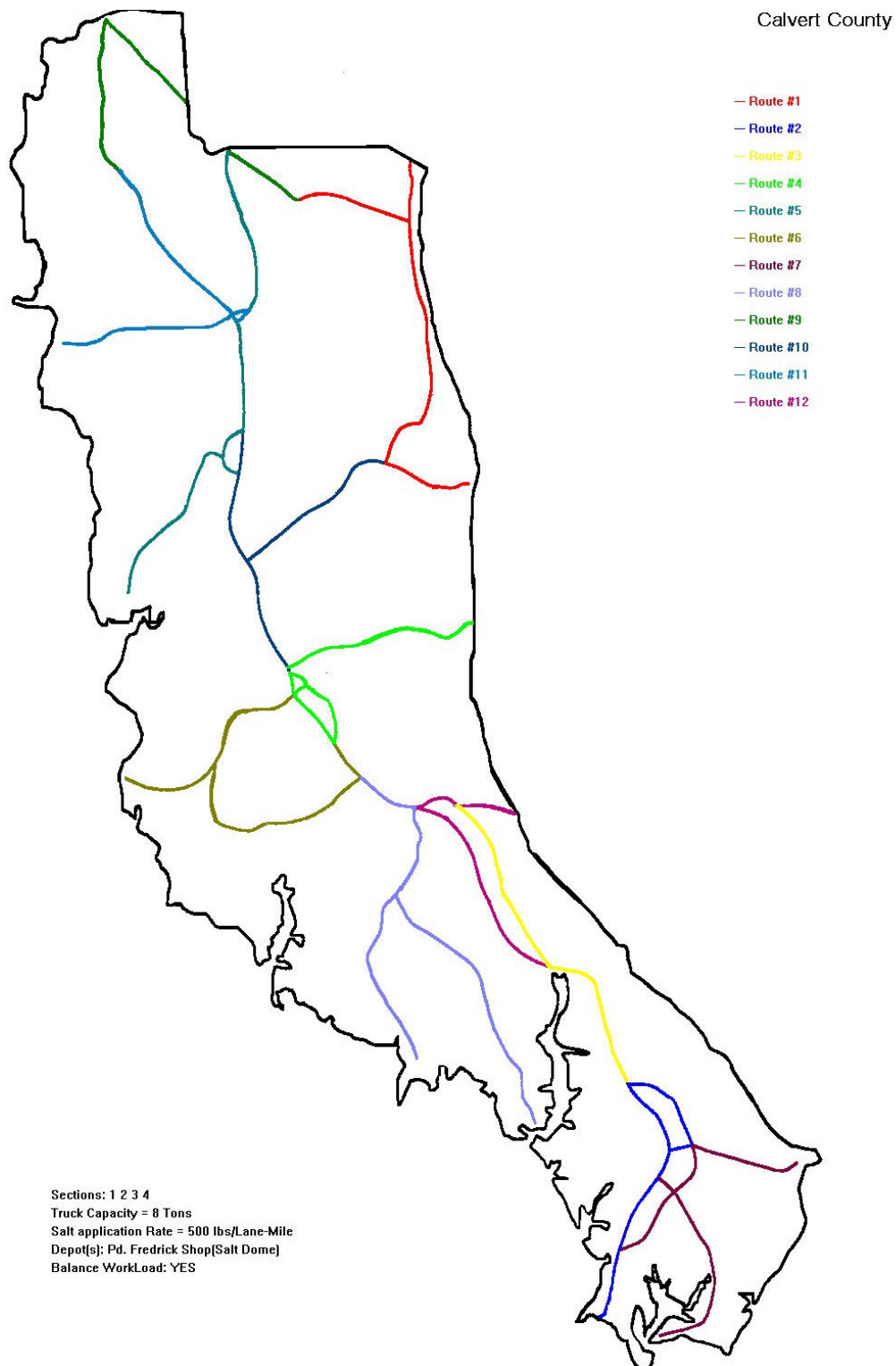


Figure 4

Output Interface for All Truck Routes

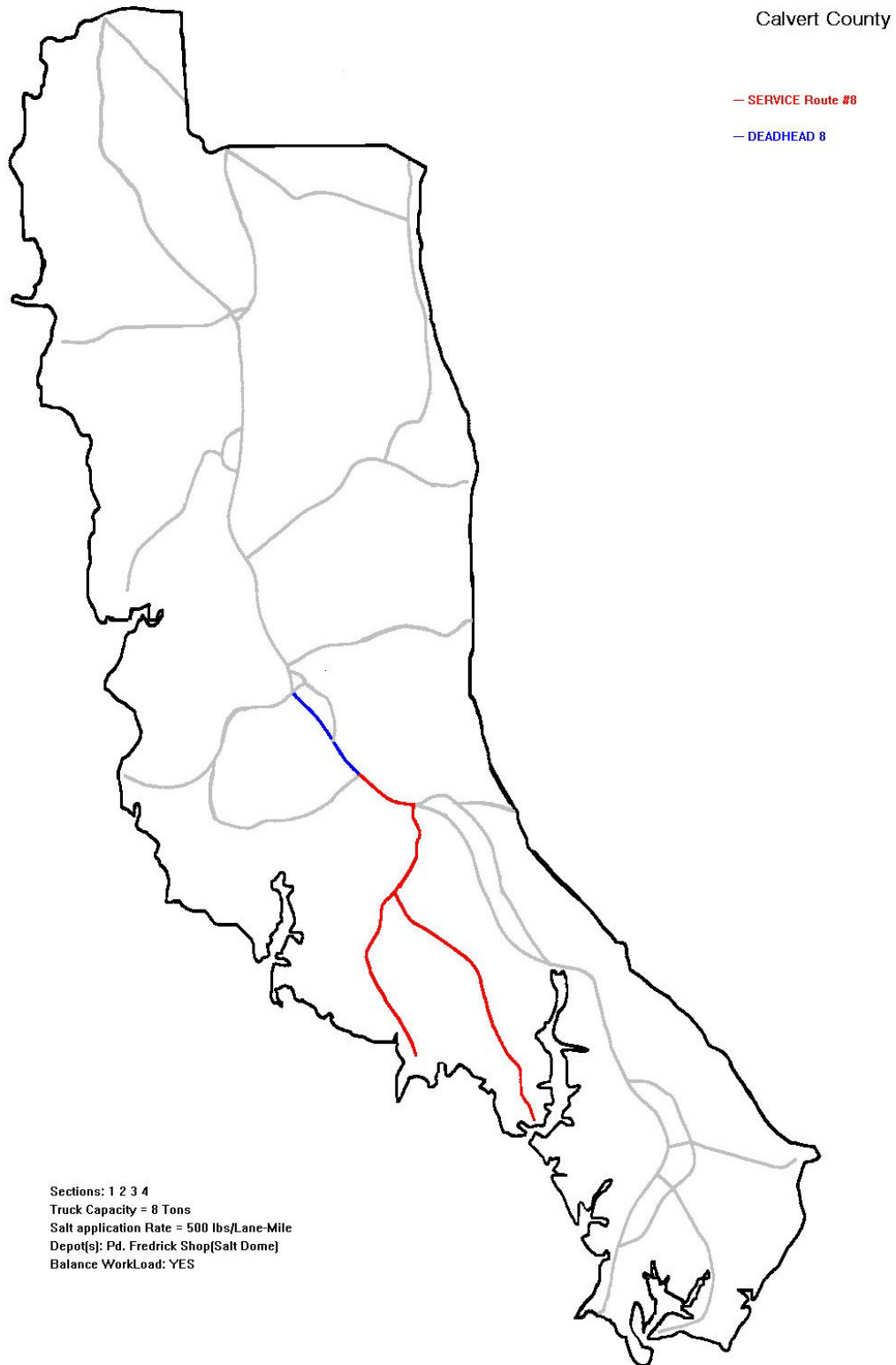


Figure 5
Output Interface for a Single Truck Route

Table 1

Detail of the routes produced by system

DEPOTS: Pd. Fredrick Shop(Salt Dome)			
ROUTE:1, Start Node	End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons Island Rd @ Dares Beach Rd	D	0.70
Solomons Island Rd @ Dares Beach Rd	Solomons Island Rd @ Plum Point Rd	D	2.77
Solomons Island Rd @ Plum Point Rd	Plum Point Rd @ Bayside Rd	D	4.09
Plum Point Rd @ Bayside Rd	Chesapeake Beach Rd @ Bayside Rd	S	6.07
Chesapeake Beach Rd @ Bayside Rd	Bayside Rd @ End of State Maint	S	0.54
Bayside Rd @ End of State Maint	Chesapeake Beach Rd @ Bayside Rd	D	0.54
Chesapeake Beach Rd @ Bayside Rd	Chesapeake Beach Rd @ Mt Harmony Rd	S	2.63
Chesapeake Beach Rd @ Mt Harmony Rd	Chesapeake Beach Rd @ Bayside Rd	S	2.63
Chesapeake Beach Rd @ Bayside Rd	Plum Point Rd @ Bayside Rd	D	6.07
Plum Point Rd @ Bayside Rd	Plum Point Rd @ End of State Maint	S	2.08
Plum Point Rd @ End of State Maint	Plum Point Rd @ Bayside Rd	D	2.08
Plum Point Rd @ Bayside Rd	Solomons Island Rd @ Plum Point Rd	D	4.09
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Dares Beach Rd	D	2.77
Solomons Island Rd @ Dares Beach Rd	Pd. Fredrick Shop(Salt Dome)	D	0.70
ROUTE:2, Start Node	End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons St @ Main St	D	1.56
Solomons St @ Main St	Solomons Island Rd @ Sixes Rd	D	0.73
Solomons Island Rd @ Sixes Rd	Solomons Island Rd @ Broomos Island Rd	D	1.64
Solomons Island Rd @ Broomos Island Rd	Solomons Island Rd @ St Leonard Rd	D	4.99
Solomons Island Rd @ St Leonard Rd	Solomons Island Rd @ HG Trueman Rd (North)	D	3.53
Solomons Island Rd @ HG Trueman Rd (North)	Solomons Island Rd @ Cove Point Rd	S	1.76
Solomons Island Rd @ Cove Point Rd	Solomons Island Rd @ Rousby Hall Rd	S	1.10
Solomons Island Rd @ Rousby Hall Rd	Solomons Island Rd @ HG Trueman Rd (South)	S	1.83
Solomons Island Rd @ HG Trueman Rd (South)	Solomons Island Rd @ Lore St	S	1.43
Solomons Island Rd @ Lore St	Solomons Island Rd @ Thomas Johnson Bridge	S	0.90
Solomons Island Rd @ Thomas Johnson Bridge	Solomons Island Rd @ Lore St	D	0.90
Solomons Island Rd @ Lore St	Solomons Island Rd @ HG Trueman Rd (South)	D	1.43
Solomons Island Rd @ HG Trueman Rd (South)	Solomons Island Rd @ Rousby Hall Rd	S	1.83
Solomons Island Rd @ Rousby Hall Rd	Solomons Island Rd @ Cove Point Rd	S	1.10
Solomons Island Rd @ Cove Point Rd	Solomons Island Rd @ HG Trueman Rd (North)	S	1.76
Solomons Island Rd @ HG Trueman Rd (North)	HG Trueman Rd @ Cove Point Road	S	2.68
HG Trueman Rd @ Cove Point Road	Solomons Island Rd @ Cove Point Rd	D	0.25
Solomons Island Rd @ Cove Point Rd	HG Trueman Rd @ Cove Point Road	S	0.25
HG Trueman Rd @ Cove Point Road	Solomons Island Rd @ HG Trueman Rd (North)	D	2.68
Solomons Island Rd @ HG Trueman Rd (North)	Solomons Island Rd @ St Leonard Rd	D	3.53
Solomons Island Rd @ St Leonard Rd	Solomons Island Rd @ Broomos Island Rd	D	4.99
Solomons Island Rd @ Broomos Island Rd	Solomons Island Rd @ Sixes Rd	D	1.64
Solomons Island Rd @ Sixes Rd	Solomons St @ Main St	D	0.73
Solomons St @ Main St	Pd. Fredrick Shop(Salt Dome)	D	1.56
ROUTE:3, Start Node	End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons St @ Main St	D	1.56
Solomons St @ Main St	Solomons Island Rd @ Sixes Rd	D	0.73
Solomons Island Rd @ Sixes Rd	Solomons Island Rd @ Broomos Island Rd	D	1.64
Solomons Island Rd @ Broomos Island Rd	St Leonard Rd @ Govenor's Run Rd	D	0.95
St Leonard Rd @ Govenor's Run Rd	St Leonard Rd @ Calvert Beach Rd	S	1.99
St Leonard Rd @ Calvert Beach Rd	Solomons Island Rd @ St Leonard Rd	S	2.13
Solomons Island Rd @ St Leonard Rd	Solomons Island Rd @ HG Trueman Rd (North)	S	3.53
Solomons Island Rd @ HG Trueman Rd (North)	Solomons Island Rd @ St Leonard Rd	S	3.53
Solomons Island Rd @ St Leonard Rd	St Leonard Rd @ Calvert Beach Rd	S	2.13
St Leonard Rd @ Calvert Beach Rd	St Leonard Rd @ Govenor's Run Rd	D	1.99
St Leonard Rd @ Govenor's Run Rd	Solomons Island Rd @ Broomos Island Rd	D	0.95
Solomons Island Rd @ Broomos Island Rd	Solomons Island Rd @ Sixes Rd	D	1.64
Solomons Island Rd @ Sixes Rd	Solomons St @ Main St	D	0.73
Solomons St @ Main St	Pd. Fredrick Shop(Salt Dome)	D	1.56
ROUTE:4, Start Node	End Node	Operation	Milage

Pd. Fredrick Shop(Salt Dome)	Church St @ Main St	S	0.33
Church St @ Main St	Solomons St @ Main St	S	1.78
Solomons St @ Main St	Church St @ Main St	D	1.78
Church St @ Main St	Pd. Fredrick Shop(Salt Dome)	D	0.33
Pd. Fredrick Shop(Salt Dome)	Solomons Island Rd @ Dares Beach Rd	S	0.70
Solomons Island Rd @ Dares Beach Rd	Dares Beach Rd @ End of State Main	S	4.33
Dares Beach Rd @ End of State Main	Solomons Island Rd @ Dares Beach Rd	D	4.33
Solomons Island Rd @ Dares Beach Rd	Church St @ Main St	D	0.37
Church St @ Main St	Solomons Island Rd @ Dares Beach Rd	S	0.37
Solomons Island Rd @ Dares Beach Rd	Pd. Fredrick Shop(Salt Dome)	S	0.70
Pd. Fredrick Shop(Salt Dome)	Solomons St @ Main St	S	1.56
Solomons St @ Main St	Pd. Fredrick Shop(Salt Dome)	S	1.56

ROUTE:5,
Start Node

End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons Island Rd @ Dares Beach Rd	D 0.70
Solomons Island Rd @ Dares Beach Rd	Solomons Island Rd @ Plum Point Rd	D 2.77
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Old Town Rd (South)	D 2.24
Solomons Island Rd @ Old Town Rd (South)	Old Town Rd @ Hunting Creek	S 0.22
Old Town Rd @ Hunting Creek	Hunting Creek Rd @ End of State Main	S 4.43
Hunting Creek Rd @ End of State Main	Old Town Rd @ Hunting Creek	D 4.43
Old Town Rd @ Hunting Creek	Solomons Island Rd @ Old Town Rd (North)	S 0.54
Solomons Island Rd @ Old Town Rd (North)	Southern MD Blvd @ Solomons Island Rd(Salt Dome)	S 2.61
Southern MD Blvd @ Solomons Island Rd(Salt Dome)	Chesapeake Beach Rd @ Solomons Island Rd	S 4.55
Chesapeake Beach Rd @ Solomons Island Rd	Southern MD Blvd @ Solomons Island Rd(Salt Dome)	D 4.55
Southern MD Blvd @ Solomons Island Rd(Salt Dome)	Solomons Island Rd @ Old Town Rd (North)	S 2.61
Solomons Island Rd @ Old Town Rd (North)	Old Town Rd @ Hunting Creek	D 0.54
Old Town Rd @ Hunting Creek	Solomons Island Rd @ Old Town Rd (South)	D 0.22
Solomons Island Rd @ Old Town Rd (South)	Solomons Island Rd @ Plum Point Rd	D 2.24
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Dares Beach Rd	D 2.77
Solomons Island Rd @ Dares Beach Rd	Pd. Fredrick Shop(Salt Dome)	D 0.70

ROUTE:6,
Start Node

End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Adelina Rd @ Hallowing Point Rd	S 2.84
Adelina Rd @ Hallowing Point Rd	Hallowing Point Rd @ County Line	S 2.70
Hallowing Point Rd @ County Line	Adelina Rd @ Hallowing Point Rd	D 2.70
Adelina Rd @ Hallowing Point Rd	Adelina Rd @ Sixes Rd	S 1.20
Adelina Rd @ Sixes Rd	Solomons Island Rd @ Sixes Rd	S 3.79
Solomons Island Rd @ Sixes Rd	Solomons St @ Main St	S 0.73
Solomons St @ Main St	Solomons Island Rd @ Sixes Rd	S 0.73
Solomons Island Rd @ Sixes Rd	Adelina Rd @ Sixes Rd	D 3.79
Adelina Rd @ Sixes Rd	Adelina Rd @ Hallowing Point Rd	D 1.20
Adelina Rd @ Hallowing Point Rd	Pd. Fredrick Shop(Salt Dome)	D 2.84

ROUTE:7,
Start Node

End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons St @ Main St	D 1.56
Solomons St @ Main St	Solomons Island Rd @ Sixes Rd	D 0.73
Solomons Island Rd @ Sixes Rd	Solomons Island Rd @ Broomos Island Rd	D 1.64
Solomons Island Rd @ Broomos Island Rd	Solomons Island Rd @ St Leonard Rd	D 4.99
Solomons Island Rd @ St Leonard Rd	Solomons Island Rd @ HG Trueman Rd (North)	D 3.53
Solomons Island Rd @ HG Trueman Rd (North)	Solomons Island Rd @ Cove Point Rd	D 1.76
Solomons Island Rd @ Cove Point Rd	HG Trueman Rd @ Cove Point Road	D 0.25
HG Trueman Rd @ Cove Point Road	Cove Point Rd @ End of State Main	S 2.43
Cove Point Rd @ End of State Main	HG Trueman Rd @ Cove Point Road	D 2.43
HG Trueman Rd @ Cove Point Road	HG Trueman Rd @ Rousby Hall Rd	S 1.39
HG Trueman Rd @ Rousby Hall Rd	Rousby Hall Rd @ End of State Main	S 4.54
Rousby Hall Rd @ End of State Main	HG Trueman Rd @ Rousby Hall Rd	D 4.54
HG Trueman Rd @ Rousby Hall Rd	Solomons Island Rd @ Rousby Hall Rd	D 0.11
Solomons Island Rd @ Rousby Hall Rd	HG Trueman Rd @ Rousby Hall Rd	S 0.11
HG Trueman Rd @ Rousby Hall Rd	Solomons Island Rd @ HG Trueman Rd (South)	S 1.84
Solomons Island Rd @ HG Trueman Rd (South)	HG Trueman Rd @ Rousby Hall Rd	D 1.84
HG Trueman Rd @ Rousby Hall Rd	HG Trueman Rd @ Cove Point Road	D 1.39
HG Trueman Rd @ Cove Point Road	Solomons Island Rd @ Cove Point Rd	D 0.25
Solomons Island Rd @ Cove Point Rd	Solomons Island Rd @ HG Trueman Rd (North)	D 1.76
Solomons Island Rd @ HG Trueman Rd (North)	Solomons Island Rd @ St Leonard Rd	D 3.53
Solomons Island Rd @ St Leonard Rd	Solomons Island Rd @ Broomos Island Rd	D 4.99
Solomons Island Rd @ Broomos Island Rd	Solomons Island Rd @ Sixes Rd	D 1.64
Solomons Island Rd @ Sixes Rd	Solomons St @ Main St	D 0.73
Solomons St @ Main St	Pd. Fredrick Shop(Salt Dome)	D 1.56

ROUTE:8, Start Node	End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons St @ Main St	D	1.56
Solomons St @ Main St	Solomons Island Rd @ Sixes Rd	D	0.73
Solomons Island Rd @ Sixes Rd	Solomons Island Rd @ Broomos Island Rd	S	1.64
Solomons Island Rd @ Broomos Island Rd	Broomes Island Rd @ Maekall Rd	S	2.24
Broomes Island Rd @ Maekall Rd	Maekall Rd @ Endof State Maint	S	1.75
Maekall Rd @ Endof State Maint	Broomes Island Rd @ Maekall Rd	D	1.75
Broomes Island Rd @ Maekall Rd	Broomes Island Rd @ End of State Maint	S	4.42
Broomes Island Rd @ End of State Maint	Broomes Island Rd @ Maekall Rd	D	4.42
Broomes Island Rd @ Maekall Rd	Solomons Island Rd @ Broomos Island Rd	D	2.24
Solomons Island Rd @ Broomos Island Rd	Solomons Island Rd @ Sixes Rd	S	1.64
Solomons Island Rd @ Sixes Rd	Solomons St @ Main St	D	0.73
Solomons St @ Main St	Pd. Fredrick Shop(Salt Dome)	D	1.56
ROUTE:9, Start Node	End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons Island Rd @ Dares Beach Rd	D	0.70
Solomons Island Rd @ Dares Beach Rd	Solomons Island Rd @ Plum Point Rd	D	2.77
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Old Town Rd (South)	D	2.24
Solomons Island Rd @ Old Town Rd (South)	Solomons Island Rd @ Old Town Rd (North)	D	0.68
Solomons Island Rd @ Old Town Rd (North)	Southern MD Blvd @ Solomons Island Rd(Salt Dome)	D	2.61
Southern MD Blvd @ Solomons Island Rd(Salt Dome)	Southern MD Blvd @ Mt Harmony Rd	D	4.32
Southern MD Blvd @ Mt Harmony Rd	Southern MD Blvd @ Chesapoake Beach Rd	S	4.54
Southern MD Blvd @ Chesapoake Beach Rd	Chesapoake Beach Rd @ Solomons Island Rd	D	4.14
Chesapoake Beach Rd @ Solomons Island Rd	Chesapoake Beach Rd @ Mt Harmony Rd	D	1.99
Chesapoake Beach Rd @ Mt Harmony Rd	Chesapoake Beach Rd @ Solomons Island Rd	S	1.99
Chesapoake Beach Rd @ Solomons Island Rd	Southern MD Blvd @ Chesapoake Beach Rd	S	4.14
Southern MD Blvd @ Chesapoake Beach Rd	Southern MD Blvd @ Mt Harmony Rd	S	4.54
Southern MD Blvd @ Mt Harmony Rd	Southern MD Blvd @ Solomons Island Rd(Salt Dome)	D	4.32
Southern MD Blvd @ Solomons Island Rd(Salt Dome)	Solomons Island Rd @ Old Town Rd (North)	D	2.61
Solomons Island Rd @ Old Town Rd (North)	Solomons Island Rd @ Old Town Rd (South)	D	0.68
Solomons Island Rd @ Old Town Rd (South)	Solomons Island Rd @ Plum Point Rd	D	2.24
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Dares Beach Rd	D	2.77
Solomons Island Rd @ Dares Beach Rd	Pd. Fredrick Shop(Salt Dome)	D	0.70
ROUTE:10, Start Node	End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons Island Rd @ Dares Beach Rd	D	0.70
Solomons Island Rd @ Dares Beach Rd	Solomons Island Rd @ Plum Point Rd	S	2.77
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Old Town Rd (South)	S	2.24
Solomons Island Rd @ Old Town Rd (South)	Solomons Island Rd @ Old Town Rd (North)	S	0.68
Solomons Island Rd @ Old Town Rd (North)	Solomons Island Rd @ Old Town Rd (South)	S	0.68
Solomons Island Rd @ Old Town Rd (South)	Solomons Island Rd @ Plum Point Rd	S	2.24
Solomons Island Rd @ Plum Point Rd	Plum Point Rd @ Bayside Rd	S	4.09
Plum Point Rd @ Bayside Rd	Solomons Island Rd @ Plum Point Rd	D	4.09
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Dares Beach Rd	S	2.77
Solomons Island Rd @ Dares Beach Rd	Pd. Fredrick Shop(Salt Dome)	D	0.70
ROUTE:11, Start Node	End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons Island Rd @ Dares Beach Rd	D	0.70
Solomons Island Rd @ Dares Beach Rd	Solomons Island Rd @ Plum Point Rd	D	2.77
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Old Town Rd (South)	D	2.24
Solomons Island Rd @ Old Town Rd (South)	Solomons Island Rd @ Old Town Rd (North)	D	0.68
Solomons Island Rd @ Old Town Rd (North)	Southern MD Blvd @ Solomons Island Rd(Salt Dome)	D	2.61
Southern MD Blvd @ Solomons Island Rd(Salt Dome)	Lower Marlboro Rd @ End of State Maint	S	4.48
Lower Marlboro Rd @ End of State Maint	Southern MD Blvd @ Solomons Island Rd(Salt Dome)	D	4.48
Southern MD Blvd @ Solomons Island Rd(Salt Dome)	Southern MD Blvd @ Mt Harmony Rd	S	4.32
Southern MD Blvd @ Mt Harmony Rd	Southern MD Blvd @ Solomons Island Rd(Salt Dome)	S	4.32
Southern MD Blvd @ Solomons Island Rd(Salt Dome)	Solomons Island Rd @ Old Town Rd (North)	D	2.61
Solomons Island Rd @ Old Town Rd (North)	Solomons Island Rd @ Old Town Rd (South)	D	0.68
Solomons Island Rd @ Old Town Rd (South)	Solomons Island Rd @ Plum Point Rd	D	2.24
Solomons Island Rd @ Plum Point Rd	Solomons Island Rd @ Dares Beach Rd	D	2.77
Solomons Island Rd @ Dares Beach Rd	Pd. Fredrick Shop(Salt Dome)	D	0.70
ROUTE:12, Start Node	End Node	Operation	Milage
Pd. Fredrick Shop(Salt Dome)	Solomons St @ Main St	D	1.56
Solomons St @ Main St	Solomons Island Rd @ Sixes Rd	D	0.73
Solomons Island Rd @ Sixes Rd	Solomons Island Rd @ Broomos Island Rd	D	1.64
Solomons Island Rd @ Broomos Island Rd	Solomons Island Rd @ St Leonard Rd	S	4.99
Solomons Island Rd @ St Leonard Rd	Solomons Island Rd @ Broomos Island Rd	S	4.99

Solomons Island Rd @ Broomos Island Rd	St Leonard Rd @ Govenor's Run Rd	S	0.95
St Leonard Rd @ Govenor's Run Rd	Govenor's Run Rd @ End of State Maint	S	0.95
Govenor's Run Rd @ End of State Maint	St Leonard Rd @ Govenor's Run Rd	D	0.95
St Leonard Rd @ Govenor's Run Rd	Solomons Island Rd @ Broomos Island Rd	D	0.95
Solomons Island Rd @ Broomos Island Rd	Solomons Island Rd @ Sixes Rd	D	1.64
Solomons Island Rd @ Sixes Rd	Solomons St @ Main St	D	0.73
Solomons St @ Main St	Pd. Fredrick Shop(Salt Dome)	D	1.56

Table 2
Route Information Summary

===== Routes Summary =====		
ROUTE : 1	Service Mile : 13.95	Deadhead Mile : 23.81
ROUTE : 2	Service Mile : 14.64	Deadhead Mile : 30.16
ROUTE : 3	Service Mile : 13.31	Deadhead Mile : 11.75
ROUTE : 4	Service Mile : 11.33	Deadhead Mile : 6.81
ROUTE : 5	Service Mile : 14.96	Deadhead Mile : 21.16
ROUTE : 6	Service Mile : 11.99	Deadhead Mile : 10.53
ROUTE : 7	Service Mile : 10.31	Deadhead Mile : 39.23
ROUTE : 8	Service Mile : 11.69	Deadhead Mile : 12.99
ROUTE : 9	Service Mile : 15.21	Deadhead Mile : 32.77
ROUTE : 10	Service Mile : 15.47	Deadhead Mile : 5.49
ROUTE : 11	Service Mile : 13.12	Deadhead Mile : 22.48
ROUTE : 12	Service Mile : 11.88	Deadhead Mile : 9.76
===== Network Summary =====		
Total Number of Trucks : 12		
Total Service Miles : 157.86		
Total Deadhead Miles : 226.94		
Minimum Workload : 10.31	for route : 7	
Maximum WorkLoad : 15.47	for route : 10	

To evaluate the results, a lower bound for the number of trucks is obtained. The lower bound is calculated by relaxing the integrality constraint from the CMST formulation and computing the linear program in CPLEX. The solution to the relaxation of CMST problem shows a lower bound of 11 trucks. Comparison of the lower bound with the solution obtained from the genetic algorithm with first fit heuristic shows that this algorithm has performed very well for this problem.

5. Conclusions and Directions for Future Research

In this research we proposed an approach for a real-world snow emergency route assignment. The proposed approach seems to be very efficient in solving problems of the size equivalent to those in Calvert County. This indicates that the algorithm can easily be implemented in a decision support system that can be used to examine what-if scenarios.

Another important aspect is that the algorithm can do the route assignment dynamically. The salting consumption rate varies with the storm characteristics and the current snow route assignment is designed using an average consumption rate. Using this procedure, one can develop different plans based on different levels of salt consumption rates.

This study indicates that the proposed procedure can be applied to real-world problems successfully. Although the problem is in a relatively small network, we still are able to have improvements over the original route design.

As an immediate and most important extension to the current system we need to develop an algorithm that incorporates different size trucks with different capacities. A second step in extensions of the current system is to implement it in another County that has a more sophisticated network. To that end certain modifications to the underlying model and algorithms would no doubt be necessary, however, the effort invested in making these

modifications will have significant return in terms of operational cost savings for the State of Maryland.

REFERENCES

- Benavent, E. V. Campos, A. Corberan, and E. Mota, 1992, "The capacitated arc routing problem: lower bounds," Networks, vol.22, no.7, p. 669-90
- Bodin, L. and L. Levy, 1991, "The Arc partitioning Problem," European Journal of Operational Research, 53, 393-401.
- Campbell, J.F., and A. Langevin, 1995, "Operations Management for Urban Snow Removal and Disposal," Transportation Research A, pp. 359-369.
- Chandy, K. M. and R.A. Russell, 1972, "The Design of Multipoint Linkages in a Teleprocessing Tree Network," IEEE Trans. Comp., vol. C-21, No. 10, pp. 1062-1066.
- Chandy, K.M. and T. Lo, 1973, "The Capacitated Minimum Spanning Tree," Networks, vol. 3, No. 2.
- Chapleau, L., J.A. Ferland, G. Lapalme and J. Rousseau, 1984, "A Parallel Insert Method for the Capacitated Arc Routing Problem," Operations Research Letters, 3, p. 95-99.
- Christofides, N, 1973, "The Optimum Traversal of a Graph," Omega 1, p. 461-470
- Clark, G. and J.W. Wright, 1964, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research, 12, p. 568-581.
- Cook, T.M., and B.S. Alprin, 1976, "Snow and Ice Removal in Urban Environment," Management Science, pp. 227-234.
- Cowen, L.J., 1994, "A Linear-Time Algorithm for Network Decomposition," DIMACS Technical Report, 94-95.
- de Cordoba, P.F. L.M. Garcia Raffi, and J.M. Sanchis, 1998, "A Heuristic Algorithm Based on Monte Carlo Methods for the Rural Postman Problem," Computers & Operations Research, vol.25, no.12, p. 1097-106
- Eglese, R.W., 1994, "Routing Winter Gritting Vehicles," Discrete Applied Mathematics, pp. 231-244.
- England, R., 1982, "Computer Analysis Ensures a Clean Sweep," Surveyor 6, May, 15.
- Gavish, B., 1982, "Topological Design of Centralized Computer Networks – formulations and algorithms," Networks, p355-377
- Gelinas, E, 1992, "Le Probleme du Postier Chinois Avec Contraintes Generales de Preseance," M. Sc. A Dissertation, Ecole Polytechnique de Montreal.

Gen, M., and R. Cheng, 2000, "Genetic Algorithms & Engineering Optimization," Wiley, New York.

Gilbert, J., 1990, "Fenerateur D'itineraires D'enlevement de la Neige," Publication CRT-648, Centre for Research on Transportation, University of Montreal.

Golden, B.L., J.S. Dearmon and E.K. Baker, 1983, "Computational Experiments with Algorithms for a Class of Routing Problem," Computers Operations Research, p. 47-59.

Golden, B.L. and R.T. Wong, 1981, "Capacitated Arc Routing Problems," Networks, p. 305-315.

Gouveia, L., 1994, "A 2n Constraint Formulation For the Capacitated Minimal Spanning Tree Problem," Networks, pp130-141

Guan, M., 1962, "Graphic Programming Using Odd and Even Points," Chinese Mathematics, 1, p. 273- 277.

Haslam, E., and J.R. Wright, 1991, "Application of Routing Technologies to Rural Snow and Ice Control," Transportation Research Record, 1304, pp. 202-211.

Kawatra, R., and D. Bricker, 2000, "A multiperiod planning model for the capacitated minimal spanning tree problem," European Journal of Operation Research 121, pp412-419

Kershenbaum, A., 1974, "Computing Capacitated Minimal Spanning Trees Efficiently," Networks, pp 299-310

Kruskal, J.B., 1956, "On the Shortest Spanning Subtree of a Graph and the Travelling Salesman Problem," Proc. of American Math. Society, vol. 7.

Larson, R.C., and A.R. Odoni, 1981, "Urban Operations Research," Prentice Hall, Englewood Cliffs, NJ.

Liebling, T.M., 1973, "Routing Problems for Street Cleaning and Snow Removal," in Models for Environmental Pollution Control, R.Deininger (ed.), pp. 363-374, Ann Arbor Publishers.

Marks, H.D., and R. Stricker, 1971, "Routing for Public Service Vehicles," ASCE Journal of the Urban Planning and Development Division 97, pp. 16-178.

Michalewicz, Z., 1999, "Genetic Algorithms + Data Structures = Evolution Progress," Springer-Verlag , New York.

Mourao, M.C., and M.T. Almeida, 2000, "Lower-bounding and heuristic methods for a refuse collection vehicle routing problem," European Journal of Operational Research, vol.121, no.2, p. 420-34

Pearn, W. L., 1989, "Approximate Solutions for the Capacitated Arc Routing Problem," Computers Operations Research, pp. 589–600.

Pearn, W. L., 1991, "Augment-Insert Algorithms for the Capacitated Arc Routing Problem," Computers Operations Research, pp. 189–198.

Prim, R.C., 1957, "Shortest Connection Networks and Some Generalizations," Bell System Technical Journal, vol 36.

Wang, J-Y., P. Kandula, and J.R. Wright, 1995, "Evaluation of Computer-Generated Routes for Improved Snow and Ice Control," Transportation Research Record 1509, p 15 – 21.

Win, Z., 1987, "Contributions to Routing Problems," Ph.D. Dissertation, University of Augsburg, Germany.

Lee, Y-H., M-J Kang, and C-G Han ,1996, "A genetic algorithm for obtaining a near-optimal solution in rural postman problem," Journal of KISS(A) (Computer Systems and Theory), vol.23, no.11, p. 1118-25

APPENDIX

CALVERT COUNTY SNOW EMERGENCY DECISION SUPPORT SYSTEM

User Manual